Nombres de Armstrong à trois chiffres [lp07] - Examen

Karine Zampieri, Stéphane Rivière

Unisciel algoprog Version 17 mai 2018

Table des matières

alg - Nombres d'Armstrong (Solution)

Mots-Clés Structures répétitives ■
Requis Structures de base, Structures conditionnelles, Structures répétitives ■
Difficulté • • ◦ (15 à 20 min) ■

Objectif
Cet exercice détermine par force brute les nombres de Armstrong à trois chiffres. (image : google/images)

...(énoncé page suivante)...

1 Nombres de Armstrong / pgarmstrong

1.1 Nombres d'Armstrong à trois chiffres

Définition
Un entier positif n à trois chiffres est un nombre de ARMSTRONG si la somme des cubes de ses chiffres est égale au nombre. Autrement dit, soit $n = \overline{cba} = 100 \times c + 10 \times b + a$ (où a, b, c sont les chiffres de n). Si :

$$n = a^3 + b^3 + c^3$$

Exemple
$153 = 1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$ est un nombre d'ARMSTRONG ainsi que 371.

1.2 Trois itératives imbriquées

Une première méthode consiste donc à utiliser **trois itératives imbriquées** pour passer en revue tous les nombres de 100 à 999 :

- Pour les chiffres de la centaine de 1 à 9
  - Pour les chiffres de la dizaine de 0 à 9
    - Pour les chiffres de l'unité de 0 à 9
      - Tester la combinaison (centaine, dizaine, unité)

Écrivez un algorithme qui recherche et affiche **tous** les entiers de ARMSTRONG compris entre 100 et 999 en utilisant **trois boucles** `Pour` **imbriquées** de compteurs `c` (centaine), `b` (dizaine) et `a` (unité).

Testez (quatre solutions).
Résultat d'exécution :

```
==> 153
==> 370
==> 371
==> 407
FIN
```

1.3 Une seule itérative

Une autre méthode consiste à passer en revue tous les nombres de 100 à 999 et à récupérer les chiffres de la centaine, la dizaine et l'unité.

Soit n un entier.

- Quelle est l'action d'une division par 10 sur n ?
- Et celle du modulo ?

Exemple : n vaut 395.

Solution simple

Une division par 10 a pour action de décaler les chiffres de n d'un cran vers la droite. Le modulo par 10 a pour action d'extraire le chiffre de droite de n. Par exemple, si n vaut 395 alors le modulo par 10 donne 5 et la division entière par 10 donne 39.

Complétez votre algorithme pour que, utilisant **une seule** répétitive, il trouve par force brute toutes les combinaisons et les affiche au fur et à mesure.

Aide détaillée

La répétitive doit passer en revue tous les entiers `n` de 100 à 999.
Dans celle-ci :

- Récupérez le chiffre des unités dans `a`, des dizaines dans `b` et celui des centaines dans `c`. Rappel : l'opération modulo par 10 permet de récupérer le dernier chiffre d'un entier et la division par 10 fait perdre son dernier chiffre.
- Testez la combinaison (centaine, dizaine, unité).

Testez. (Les mêmes quatre solutions ci-dessus.)

1.4 Validation

Validez votre algorithme avec la solution.

Solution alg @[pgarmstrong.alg]

```
Algorithme pgfbserrure
Variable a , b , c : Entier
Variable n , n3 : Entier
Début
  |  Pour c <- 1 à 9 Faire
  |    |  Pour b <- 0 à 9 Faire
  |    |    |  Pour a <- 0 à 9 Faire
  |    |    |    |  n <- c * 100 + b * 10 + a
  |    |    |    |  n3 <- c*c*c + b*b*b + a*a*a
  |    |    |    |  Si ( n = n3 ) Alors
  |    |    |    |    |  Afficher ( "==> " , c , b , a )
  |    |    |    |  FinSi
  |    |    |  FinPour
  |    |  FinPour
  |  FinPour
  |  Afficher ( "FIN" )
  |  Pour n <- 100 à 999 Faire
```

```
  |    |  a <- Modulo ( n , 10 )
  |    |  b <- Modulo ( DivEnt ( n , 10 ) , 10 )
  |    |  c <- Modulo ( DivEnt ( n , 100 ) , 10 )
  |    |  n3 <- c*c*c + b*b*b + a*a*a
  |    |  Si ( n = n3 ) Alors
  |    |    |  Afficher ( "==> " , n )
  |    |  FinSi
  |  FinPour
  |  Afficher ( "FIN" )
Fin
```

2 Nombre de Armstrong (6 points)

Objectif
Un nombre de Amstrong est un entier positif dont la somme des cubes des chiffres vaut cet entier. Exemple : $153 = 1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$. Déterminer les nombres de Amstrong à trois chiffres.

(3 points) Écrivez un algorithme qui calcule et affiche tous les entiers de Amstrong compris entre 100 et 999 en utilisant **trois** boucles `Pour` **imbriquées**.

Testez. (voir plus bas)

(2 points) Faites de même en utilisant **une seule** boucle `Pour`.

(1 point) Dans une des deux boucles de recherche, ajoutez les instructions qui comptent le nombre d'entiers vérifiant la relation puis affichez-le.

Testez. Résultat d'exécution :

```
==> 039
==> 066
==> 093
==> 309
==> 336
==> 363
==> 390
==> 606
==> 633
==> 660
==> 903
==> 930
FIN
```

Validez votre algorithme avec la solution.

3 Références générales

Comprend [Felea-PG1 :c3 :ex51], [Rohaut-JV1 :c4 :xm], [Rousselet-PY1 :c8 :ex8] ■